

DATA PROCESSING METHOD, RECORDING MEDIUM AND DATA PROCESSING DEVICE

Publication number: JP2002008060

Publication date: 2002-01-11

Inventor: SAITO YASUHIKO; UMINAGA MASAHIRO; YAMADA KOSHI

Applicant: HITACHI LTD

Classification:

- international: **G06T17/40; G06K9/48; G06T3/00; G06T9/00; H03M7/24; H03M7/30; G06T17/40; G06K9/48; G06T3/00; G06T9/00; H03M7/14; H03M7/30; (IPC1-7): G06T17/40; G06T3/00**

- European: G06K9/48; G06T9/00F; H03M7/24; H03M7/30

Application number: JP20000194935 20000623

Priority number(s): JP20000194935 20000623

Also published as:



US6873324 (B2)



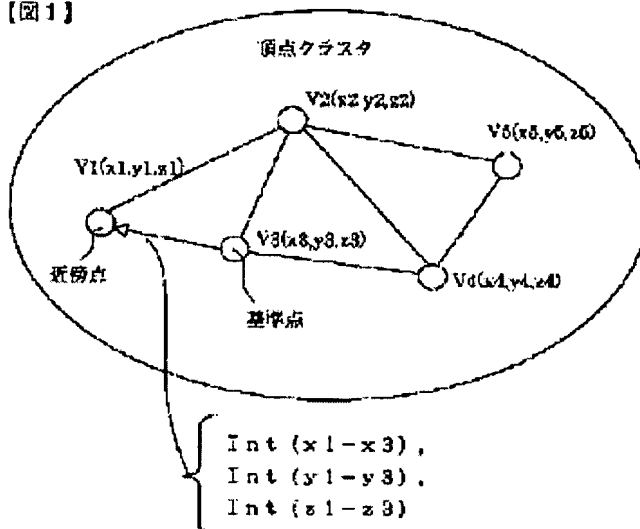
US2001055032 (A1)

[Report a data error here](#)

Abstract of JP2002008060

PROBLEM TO BE SOLVED: To provide a data compressing and extending method capable of easily compressing the vertex data of a polygon and capable of extending the compressed data without giving an excessive load onto a processor. **SOLUTION:** Vertexes of plural polygons approximating to a three dimensional shape of an image are divided into plural vertex aggregations (V1-V5), and a reference point (V3) is decided for every vertex aggregation, and the differential data between a coordinate of the vertex and the coordinates of the reference point (Int(x1-x3), Int(y1-y3), Int(z1-z3)) are obtained for the positional information of each vertex for every vertex aggregation as an integer showing the scale in relation to the predetermined scale value. The coordinate data of the reference point for every vertex aggregation and the differential data are compressed data, and the distance between each vertex and the reference point thereof is sufficiently small, and each component can be expressed with an integer. The compressed data structure has regulations at each vertex, and can be extended by converting the integer type data into the floating decimal number type data and dividing with the scale value.

【図1】



Data supplied from the **esp@cenet** database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-8060

(P2002-8060A)

(43) 公開日 平成14年1月11日 (2002.1.11)

(51) Int.Cl. ⁷	識別記号	F I	データコード* (参考)
G 0 6 T 17/40		G 0 6 T 17/40	A 5 B 0 5 0
3/00	1 0 0	3/00	1 0 0 5 B 0 5 7

審査請求 未請求 請求項の数16 O L (全 20 頁)

(21) 出願番号 特願2000-194935 (P2000-194935)

(22) 出願日 平成12年6月23日 (2000.6.23)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 斎藤 靖彦

東京都小平市上水本町五丁目20番1号 株

式会社日立製作所半導体グループ内

(73) 発明者 海永 正博

東京都小平市上水本町五丁目20番1号 株

式会社日立製作所半導体グループ内

(74) 代理人 100089071

弁理士 玉村 静世

最終頁に続く

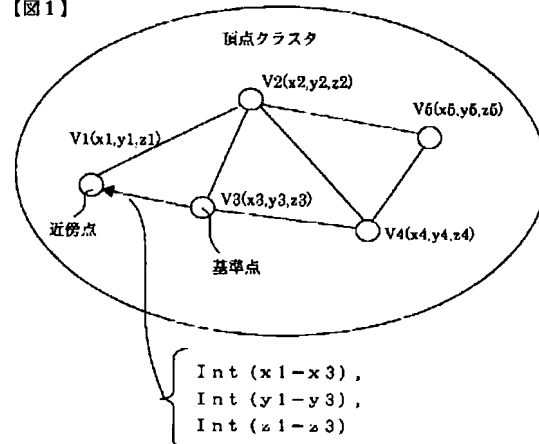
(54) 【発明の名称】 データ処理方法、記録媒体及びデータ処理装置

(57) 【要約】

【課題】 ポリゴンの頂点データに対する圧縮を簡易に行え、圧縮データに対する伸長処理もプロセッサに過大な負荷を与えないデータ圧縮・伸張方法を提供する。

【解決手段】 画像の3次元形状を近似する複数のポリゴンの頂点を複数の頂点集合 (V1~V5) に分割し、頂点集合毎に基準点 (V3) を決め、頂点集合毎にその各頂点の位置情報として、当該頂点の座標と基準点の座標との差分データ (Int (x1-x3), Int (y1-y3), Int (z1-z3)) を、所定のスケール値に対する大きさを示す整数値として求める。頂点集合毎の基準点の座標データ及び差分データが圧縮データであり、各頂点とその基準点との距離は十分小さく、各成分を整数値で表現できる。圧縮されたデータ構造は各頂点について規則的であり、整数型データから浮動小数点数型データへの変換とスケール値との乗算で伸張可能である。

【図1】



【特許請求の範囲】

【請求項1】 画像の3次元形状を近似する複数のポリゴンの頂点座標データを圧縮するデータ処理方法であって、

画像の3次元形状を近似する複数のポリゴンの頂点を複数の頂点集合に分割する第1ステップと、

頂点集合毎に基準点を定める第2ステップと、

頂点集合毎にその各頂点の位置情報として当該頂点の座標と基準点の座標との差分データを求める第3ステップと、

頂点集合毎における基準点の座標データ及び差分データを圧縮されたデータとして格納する第4ステップと、を含むことを特徴とするデータ処理方法。

【請求項2】 前記頂点の座標及び基準点の座標は浮動小数点データで与えられ、前記第3ステップは前記差分データとして、頂点の座標と基準点の座標との差分を所定のスケール値に対する大きさを示す整数値として求めることを特徴とする請求項1記載のデータ処理方法。

【請求項3】 前記第2ステップは、頂点集合毎に当該頂点集合に含まれる各頂点の座標から求められる重心点に近い頂点、又は前記重心点を、当該頂点集合の基準点として採用することを特徴とする請求項1又は2記載のデータ処理方法。

【請求項4】 コンピュータ読取り可能に画像データを記録した記録媒体であって、

前記画像データは、画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点の座標データと、夫々の頂点の座標と基準点の座標との差分データとを有して成るものであることを特徴とする記録媒体。

【請求項5】 前記基準点の座標データは浮動小数点データであり、前記差分データは所定のスケール値に対する大きさを示す整数値データであることを特徴とする請求項4記載の記録媒体。

【請求項6】 前記基準点は、前記複数のポリゴンの頂点の集合に含まれる各頂点の座標から求められる重心点の座標データ、又はその重心点に近い頂点の座標データであることを特徴とする請求項4又は5記載の記録媒体。

【請求項7】 画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と基準点の座標との差分を整数で与える差分データと、差分データの大きさの基準を浮動小数点数で与えるスケール値データとを入力する第1ステップと、

入力した前記差分データを浮動小数点数に型変換する第2ステップと、

型変換された差分データに前記スケール値データを乗算する第3ステップと、

前記第3ステップによる乗算結果を、伸張された差分データとして格納する第4ステップと、を含むことを特徴

とするデータ処理方法。

【請求項8】 画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と基準点の座標との差分を整数で与える差分データと、差分データの大きさの基準を浮動小数点数で与えるスケール値データとを格納可能な記憶手段と、

前記記憶手段から読み出された前記差分データを入力して浮動小数点数に型変換し、型変換された差分データに前記スケール値データを乗算し、乗算結果を伸張された差分データとして出力する演算制御手段と、を含んで成るものであることを特徴とするデータ処理装置。

【請求項9】 前記演算手段はCPUであり、CPUは整数データの浮動小数点数データへの型変換を指示する命令を実行可能であり、この型変換を指示する命令は、整数データのビット長を指示するビット長情報フィールドを有して成るものであることを特徴とする請求項8記載のデータ処理装置。

【請求項10】 画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と基準点の座標との差分を浮動小数点数で与える差分データとを入力し、入力した前記基準点の座標データに対してアフィン変換を行うとき、当該基準点を共有する座標データに対して当該アフィン変換から平行移動を除外した線形変換を行うことを特徴とするデータ処理方法。

【請求項11】 コンピュータ読取り可能にプログラムを記録した記録媒体であって、

前記プログラムは、コンピュータに、画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と基準点の座標との差分を浮動小数点数で与える差分データとを入力させ、入力させた前記基準点の座標データに対してアフィン変換を行う場合には、当該基準点を共有する座標データに対して当該アフィン変換から平行移動を除外した線形変換を実行させること特徴とする記録媒体。

【請求項12】 画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と基準点の座標との差分を浮動小数点数で与える差分データとを格納可能な記憶手段と、

前記記憶手段から読み出された前記基準点の座標データに対してアフィン変換を行うとき、当該基準点を共有する座標データに対して当該アフィン変換から平行移動を除外した線形変換を行う演算手段と、を含んで成るものであることを特徴とするデータ処理装置。

【請求項13】 画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と対応する基準点

の座標との差分を浮動小数点数で与える差分データとを入力し、前記複数のポリゴンの頂点に対する透視変換を行うとき、

基準点に対しては幾何学の相似の関係を用いた除算による透視変換を行って透視変換後の座標データを求め、基準点以外の頂点に対しては、マクローリン級数展開による近似計算に対応する基準点の透視変換後の座標データ及び対応する前記差分データを用いて透視変換後の座標データを求める、ことを特徴とするデータ処理方法。

【請求項14】 上記透視変換の近似計算において、透視変換の座標原点から透視面までの距離が既定値よりも小さいときは、基準点以外の頂点の座標データに対しても幾何学の相似の関係を用いた除算による透視変換を行うことを特徴とする請求項13記載のデータ処理方法。

【請求項15】 コンピュータ読取り可能にプログラムを記録した記録媒体であって、

前記プログラムは、コンピュータに、画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と対応する基準点の座標との差分を浮動小数点数で与える差分データとを入力させ、前記複数のポリゴンの頂点に対する透視変換を行うとき、入力した基準点の座標データに対しては幾何学の相似の関係を用いた除算による透視変換を行って透視変換後の座標データを取得させ、基準点以外の入力した頂点の座標データに対しては、対応する基準点の透視変換後の座標データをマクローリン級数展開による近似計算に用いて透視変換後の座標データを取得させることを特徴とする記録媒体。

【請求項16】 画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と対応する基準点の座標との差分を浮動小数点数で与える差分データとを格納可能な記憶手段と、

前記複数のポリゴンの頂点に対する透視変換を行うとき、記憶手段に格納された基準点の座標データに対しては幾何学の相似の関係を用いた除算による透視変換を行って透視変換後の座標データを求め、記憶手段に格納されたところの基準点以外の頂点の座標データに対しては、透視変換の透視面から基準点までの距離が既定値よりも大きいとき、対応する基準点の透視変換後の座標データをマクローリン級数展開による近似計算に用いて透視変換後の座標データを求め、透視変換の透視面から基準点までの距離が既定値よりも小さいとき、記憶手段に格納されたところの基準点以外の頂点の座標データに対しても幾何学の相似の関係を用いた除算による透視変換を行って透視変換後の座標データを求める演算手段と、を含んで成るものであることを特徴とするデータ処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、3次元グラフィックスデータ等の画像データに対するデータの圧縮・伸長技術、アフィン変換や透視変換などの画像処理方法、そして画像データやプログラムの記録媒体に関する。

【0002】

【従来の技術】ゲームや映画等の3次元グラフィックスを利用する分野では高精細で高速な画像処理へのニーズは年々高まる傾向にあり、画像処理装置に必要な演算量および扱うデータ量は年々増加している。

【0003】これら3次元グラフィックス分野では3次元物体をポリゴンと呼ばれる一種の多角形の集まりで表現し、それらに対して画像処理を適用する。ポリゴンを対象とした画像処理の参考文献としては、Addison Wesley社から1982年に出版された文献「(J. D. Foley & A. Van Dam著) Fundamentals of Interactive Computer Graphics」がある。

【0004】ポリゴンで表現された物体を高精細且つ滑らかに表現する方式としては物体を構成するポリゴンをより細かくし、必要なポリゴン数を増やすということが考えられる。しかし、ポリゴン数を増やすことにより、データ量および演算量が増大する。

【0005】演算量の増大に対してはプロセッサの処理能力を向上させることで対応できるが、データ量の増大に対しては画像データを記録している記録媒体の大容量化や、メモリとプロセッサのデータ転送幅(並列転送データビット数)の拡大等が必要になる。しかし、プロセッサのデータ処理能力と、外部メモリ・プロセッサ間のバスを介するデータ転送能力とのギャップは広がる傾向にあり、プロセッサの処理能力が十分であっても、バスを介するデータ転送能力がプロセッサに追従できず、全体的なデータ処理性能をデータ転送能力が律則してしまうことがある。

【0006】従来技術として、プロセッサで処理されるデータを予め圧縮し、メモリや記録媒体に格納しておき、プロセッサで処理する直前に圧縮されたデータを伸長する方法が知られている。3次元画像データ、特にポリゴンデータの圧縮データ構造としては、例えば、「J A V A (登録商標) 3D」が定める圧縮方式(SUN MICROSYSTEMS社の公開ホームページ「<http://java.sun.com/product/s/java-media/3D>」内の公開資料:「J a v a 3 D A P I S p e c i f i c a t i o n (V e r s i o n 1.1.2, J u n e 1999)」の付録Bにある3次元幾何圧縮(3D Geometry Compression))が知られている。

【0007】前記「J a v a 3 D」では、ポリゴンを構成する個々の頂点に対応してある種の命令を対応させる。これは可変長命令であり、頂点の座標、色、法線などのデータをそれ以前に処理された点に対する差分値で

持ち、さらに全体を可変長圧縮したものである。これによりネットワーク通信時の転送時間の短縮を図ることができる。

【0008】

【発明が解決しようとする課題】しかしながら、前記「Java 3D」の圧縮方式だと、圧縮された情報を伸長するには、上記命令を逐次的に解釈して伸長しながらポリゴンを再構成しなければならない。可変長圧縮データの伸長を逐次的に行う必要があるため、伸長処理が遅い傾向にある。

【0009】また、特開平11-259680号公報記載のように、時間的に連続する2面の画像データの比較において基準画像とのポリゴン差分データを作成し、基準画像データに対する差分データを持つことで、データ量を削減する技術がある。これは、画像の動きを記述するデータ量を削減するものであり、画像の静的なデータ量を削減するものではない。

【0010】本発明の目的は、ポリゴンの頂点データに対する圧縮を簡易に行うことができ、圧縮データに対する伸長処理でもプロセッサ等に過大な負荷を与えることのないデータ圧縮方法及びデータ伸張方法を提供することにある。

【0011】本発明の別の目的は、ポリゴンの頂点データに対するデータ圧縮構造との関連においてアフィン変換や透視変換などの3次元画像処理の高速化に寄与できるデータ処理方法を提供することにある。

【0012】本発明の更に別の目的は、前記データ圧縮方法及びデータ伸張方法、更には前記アフィン変換や透視変換などの高速化に寄与するデータ処理方法を容易に実現可能にするプログラムの記録媒体を提供することにある。

【0013】本発明の更に別の目的はアフィン変換や透視変換などの3次元画像処理の高速化にとって望ましいところの、ポリゴンのデータ構造或いはそのデータを記録した記録媒体を提供することにある。

【0014】本発明のその他の目的は、ポリゴンの頂点データに対するデータ圧縮構造との関連においてアフィン変換や透視変換などの3次元画像処理の高速化に寄与できるデータ処理装置を提供することにある。

【0015】本発明の前記並びにその他の目的と新規な特徴は本明細書の記述及び添付図面から明らかになるであろう。

【0016】

【課題を解決するための手段】本願において開示される発明のうち代表的なものの概要を簡単に説明すれば下記の通りである。

【0017】〔1〕《圧縮方法》

ポリゴンデータの圧縮データ構造を次のようにする。まず、複雑な3次元の物体を、より単純な剛体のような部品の集まりに分解する。例えば人体をポリゴンで表現す

る場合、顔や腕、肘などのパーツに分解する。これらパーツ毎に含まれる複数のポリゴンを構成する頂点の集合をクラスと呼ぶことにする。ポリゴン数を増やせば、各パーツを構成する複数の頂点は互いに空間的に近い範囲に存在することになる。これに着目し、空間的に互いに近い頂点の集合毎に基準点を定め、各頂点とその基準点との相対ベクトルを考えたとき、その距離が小さいため、各成分を例えば1バイトのような整数値で表現することが可能になる。従来、ポリゴンの幾何情報は、一つの原点に対して頂点座標の各座標値を4バイトの浮動小数点で表現していたことと比較すれば、データ量を約4分の1に圧縮することができる。

【0018】そのようなデータ圧縮に着目したデータ処理方法は、画像の3次元形状を近似する複数のポリゴンの頂点座標データを圧縮する方法であって、画像の3次元形状を近似する複数のポリゴンの頂点を複数の頂点集合(V1~V5)に分割する第1ステップと、頂点集合毎に基準点(V3, P)を決める第2ステップと、頂点集合毎にその各頂点の位置情報として当該頂点の座標と基準点の座標との差分データを求める第3ステップと、頂点集合毎における基準点の座標データ及び差分データを圧縮されたデータとして格納する第4ステップと、を含む。

【0019】前記頂点の座標及び基準点の座標が浮動小数点データで与えられても、前記第3ステップでは前記差分データとして、頂点の座標と基準点の座標との差分を所定のスケール値に対する大きさを示す整数値として求めることが可能である。例えば、差分データを符号付き1バイトの整数(値0~値127)で表現しようとするなら、基準点と頂点との距離の最大値を127で除した商をスケール値し、頂点座標と基準点座標との差分値を前記スケール値で除した商を差分データとすればよい。

【0020】前記頂点集合の基準点を、頂点集合毎に当該頂点集合に含まれる各頂点の座標から求められる重心点に近い頂点、又は前記重心点としてよい。これにより、頂点と相互に近い位置の基準点を容易に決めることができる。

【0021】〔2〕《伸張》

前記方法で圧縮されたデータを伸張するのに以下の手段を採用する。前記圧縮されたデータ構造は各頂点について規則的な構造をもち、例えば整数型データから浮動小数点型データへの型変換処理と、前記スケール値との乗算処理とによって、伸張処理が可能である。この伸張処理方法は簡易且つ高速処理可能であり、専用の伸張装置も低コストで済む。更に、プロセッサに比較的簡易な伸長命令を定義すれば、プロセッサの処理性能に見合う速度でデータ伸張が可能になる。

【0022】更に詳しくは、伸張を行うデータ処理方法は、画像の3次元形状を近似する複数のポリゴンの頂点

の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と基準点の座標との差分を整数で与える差分データと、差分データの大きさの基準を浮動小数点数で与えるスケール値データとを入力する第1ステップと、入力した前記差分データを浮動小数点数に型変換する第2ステップと、型変換された差分データに前記スケール値データを乗算する第3ステップと、前記第3ステップによる乗算結果を、伸張された差分データとして格納する第4ステップと、を含む。

【0023】そのようなデータ伸張方法を実現する観点のデータ処理装置は、画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と基準点の座標との差分を整数で与える差分データと、差分データの大きさの基準を浮動小数点数で与えるスケール値データとを格納可能な記憶手段を有する。そして、前記記憶手段から読み出された前記差分データを入力して浮動小数点数に型変換し、型変換された差分データに前記スケール値データを乗算し、乗算結果を伸張された差分データとして出力する演算制御手段を備える。

【0024】前記演算手段は例えばCPUであり、CPUは整数データの浮動小数点数データへの型変換を指示する命令を実行可能であり、この型変換を指示する命令は、整数データのビット長を指示するビット長情報フィールドを有している。前記型変換では整数データのビット長と所定の浮動小数点フォーマットの仮数のビット長との相違に応じたビット長拡張処理が行われるが、それに必要な整数データのビット長情報を、前記命令のデコード結果から得られるから、ビット長の異なる整数データが混在する場合も、1命令で前記型変換を伴う処理を行うことができる。処理対象とする整数データのビット長情報を制御レジスタで指定する場合に比べ、処理対象とする整数データのビット長が変わっても、その都度、余計なレジスタアクセス動作を行わずに済み、データ処理能率も向上する。

【0025】〔3〕《データ記録媒体》

前記差分データによって圧縮された画像データはコンピュータ読取り可能な記録媒体に記録して提供することができる。例えば、そのような記録媒体に記録される画像データは、画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点の座標データと、夫々の頂点の座標と基準点の座標との差分データとを有する。前記基準点の座標データは浮動小数点データであり、前記差分データは所定の標準値に対する大きさを示す整数値データである。前記基準点は、前記複数のポリゴンの頂点の集合に含まれる各頂点の座標から求められる重心点の座標データ、又はその重心点に近い頂点の座標データである。

【0026】前記差分データによって圧縮された画像データを記録媒体に記録して提供すれば、そのデータを用

いて画像処理が行なわれることにより、圧縮データの伸長がプロセッサの画像処理性能を落とさず、また、記録媒体及びデータ転送のコストを減少させることを、可能にする。

【0027】〔4〕《アフィン変換》

伸長されたデータ構造を活用した3次元画像処理の高速化のために、伸長された各頂点のデータは、基準点からの差分ベクトルで表され、各成分は例えば夫々4バイトの浮動小数点型データとされる。クラスタを構成する頂点の「アフィン変換」では基準点に対する各頂点の差分ベクトルに平行移動を除いた線形変換を適用することで、演算量の軽減が可能になる。

【0028】更に詳しくは、そのようなアフィン変換を行うデータ処理方法は、画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と基準点の座標との差分を浮動小数点数で与える差分データとを入力する。差分データは前記圧縮データ構造に対して伸張されたようなデータを想定する。そして、入力した前記基準点の座標データに対してアフィン変換を行うとき、当該基準点を共有する座標データに対して当該アフィン変換から平行移動を除外した線形変換を行う。

【0029】そのようなデータ処理方法の実現を支援するプログラムはコンピュータ読取り可能に記録媒体に記録して提供することができる。前記プログラムは、コンピュータに、画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と基準点の座標との差分を浮動小数点数で与える差分データとを入力させ、入力させた前記基準点の座標データに対してアフィン変換を行う場合には、当該基準点を共有する座標データに対して当該アフィン変換から平行移動を除外した線形変換を実行させる。

【0030】そのようなデータ処理方法を適用したデータ処理装置は、画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と基準点の座標との差分を浮動小数点数で与える差分データとを格納可能な記憶手段と、前記記憶手段から読み出された前記基準点の座標データに対してアフィン変換を行うとき、当該基準点を共有する座標データに対して当該アフィン変換から平行移動を除外した線形変換を行う演算手段と、を含んで成る。

【0031】〔5〕《透視変換》

伸長されたデータ構造を活用した3次元画像処理の高速化のために、透視変換の近似処理を行なう。すなわち、基準点を共有する頂点の集合に含まれる頂点は空間的に互いに近く存在していることから、そのような各頂点の透視変換では、基準点に対応する透視変換結果を用いて乗算と加算の組み合わせの演算で近似させて、透視変換

で必要となる除算処理を軽減できるようにする。乗算や加算に比べて除算処理に時間がかかるようなプロセッサによる処理に効果的である。

【0032】更に詳しくは、そのような透視変換を行うデータ処理方法は、画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と対応する基準点の座標との差分を浮動小数点数で与える差分データとを入力し、前記複数のポリゴンの頂点に対する透視変換を行うとき、基準点に対しては幾何学の相似の関係をを用いた除算による透視変換を行って透視変換後の座標データを求める。基準点以外の頂点に対しては、マクローリン級数展開による近似計算に対応する基準点の透視変換後の座標データ及び対応する差分データを用いて透視変換後の座標データを求める。

【0033】上記透視変換の近似計算において、透視変換の座標原点から透視面までの距離が既定値よりも小さいときは、近似計算による誤差が無視し得ないときである、その場合には、基準点以外の頂点の座標データに対しても幾何学の相似の関係をを用いた除算による透視変換を行う方がよい。

【0034】そのようなデータ処理方法の実現を支援するプログラムはコンピュータ読取り可能に記録媒体に記録して提供することができる。前記プログラムは、コンピュータに、画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と対応する基準点の座標との差分を浮動小数点数で与える差分データとを入力させ、前記複数のポリゴンの頂点に対する透視変換を行うとき、入力した基準点の座標データに対しては幾何学の相似の関係をを用いた除算による透視変換を行って透視変換後の座標データを取得させ、基準点以外の入力した頂点の座標データに対しては、対応する基準点の透視変換後の座標データをマクローリン級数展開による近似計算に用いて透視変換後の座標データを取得させる。

【0035】そのようなデータ処理方法を適用したデータ処理装置は、画像の3次元形状を近似する複数のポリゴンの頂点の集合に対する基準点を浮動小数点数で与える座標データと、夫々の頂点の座標と対応する基準点の座標との差分を浮動小数点数で与える差分データとを格納可能な記憶手段を有する。更に、前記複数のポリゴンの頂点に対する透視変換を行うとき、記憶手段に格納された基準点の座標データに対しては幾何学の相似の関係をを用いた除算による透視変換を行って透視変換後の座標データを求め、記憶手段に格納されたところの基準点以外の頂点の座標データに対しては、透視変換の透視面から基準点までの距離が既定値よりも大きいとき、対応する基準点の透視変換後の座標データをマクローリン級数展開による近似計算に用いて透視変換後の座標データを求め、透視変換の透視面から基準点までの距離が既定値

よりも小さいとき、記憶手段に格納されたところの基準点以外の頂点の座標データに対しても幾何学の相似の関係をを用いた除算による透視変換を行って透視変換後の座標データを求める演算手段を備える。

【0036】

【発明の実施の形態】《画像処理システム》図4は本発明の差分データ構造を活用する画像処理システムのブロック図である。記録媒体2には画像処理対象とされる画像データ、そしてアプリケーションプログラムとしての画像処理プログラムが格納されている。記録媒体2は例えばフロッピー（登録商標）ディスク、CD-ROMディスク、MO（光磁気）ディスク、フラッシュメモリーカード等から成る。記録媒体2に対する画像データや画像処理プログラムの記録は磁氣的、電氣的、光学的、ディスクの樹脂材料に対する機械的塑性変形などを介して行なわれている。記録媒体2に記録されている画像データ及び画像処理プログラムは画像処理システム1によって読み取り可能にされる。画像処理システム1は、ワークステーション又はパーソナルコンピュータなどのコンピュータによって構成される。記録媒体2に記録されている画像データ及び画像処理プログラムは、プロセッサ5の制御により、当該記録媒体2に対応したディスクドライブ装置3を介して読み取られる。読み取られた画像データは差分データ変換器4に供給される。差分データ変換器4は記録媒体2から読み出された画像データをプロセッサ5にとって画像処理し易いデータ構造に変換する機能を有する。記録媒体2から読み取られた画像処理プログラムはプロセッサ5の制御でメモリ7にロードされ、プロセッサ5によって実行される。プロセッサ5はOS（オペレーティングシステム）を実行して基本的なデータ処理を行う。さらにプロセッサ5はキーボードやポインティングデバイスなどの入力装置6からの指示を参照して画像処理プログラムを実行することにより画像データを処理する。処理した画像データの表示は画像表示装置8で行う。

【0037】図4に例示される画像処理システムを用いて例えば3次元画像処理対象とされる画像データは、空間内の頂点の集まりから成り、各頂点は、3次元の座標値、さらに明るさを示すための頂点法線データ（法線ベクトル）等の属性データを持っている。プロセッサ5はこれら座標値や属性データに対してアフィン変換、クリッピング、シェーディング、透視変換、及びレンダリングなどの処理をパイプライン処理的に行う。このような一連の処理はグラフィックパイプラインと呼ばれることもある。以下の説明では、グラフィックパイプライン処理のうち、本発明の差分データ構造を利用したアフィン変換及び透視変換に関するデータ処理について述べる。

【0038】《整数差分データ》図1には3次元画像を構成する頂点集合の一例を示す。ここでは3角形の集合により3次元空間内の物体表面を表す。このような三角

形をポリゴンと呼ぶ。図1では3つの頂点V1、V2、V3を持つポリゴンをポリゴンV1V2V3と表記することにすれば、図1はポリゴンV1V2V3、ポリゴンV2V3V4、ポリゴンV3V4V5をもつ物体表面を表している。このような幾何情報は頂点の並びV1、V2、V3、V4、V5をこの順番で与えることにより表現できる。すなわち最初の2つの頂点V1、V2は例外として、次の頂点V3にポリゴンV1V2V3を対応させ、次の頂点V4と直前のポリゴンを構成する頂点のうち、順番の新しい2つ、すなわち頂点V2、V3とによりポリゴンV2V3V4が表現できる。同様に次の頂点V5によりポリゴンV3V4V5が表現される。このように頂点を1つ加える毎に新しいポリゴンを構成するデータ表現をストリップ表現と呼ぶ。各頂点の位置はその3次元座標値、すなわちx座標値、y座標値、z座標値で確定する。各座標値は浮動小数点数で表現されるものとする。例えば、頂点V1のx座標値、y座標値、z座標値はそれぞれx1、y1、z1で表し、各座標値は浮動小数点数とする。各頂点は、さらに3次元座標値以外にも、色属性や、法線ベクトル情報等の属性データも持っている。

【0039】図1では、例として5つの頂点V1、V2、V3、V4、V5の集まりを示している。このように予め決められた頂点の集まりをクラスタと呼ぶことにする。クラスタは複雑な3次元物体を構成する頂点集合をより単純ないくつかの部分集合に分割したときの頂点集合を表現するのに使用する。これらクラスタの例としては、剛体的な特性、すなわち空間内の移動でその形を変えないような特性をもつ物体を構成する頂点の集まりなどがある。

【0040】図2には上記クラスタデータのデータ構造が例示される。図2の(A)に例示されるようにクラスタデータ24は、そのクラスタに対応するヘッダデータ10及びそのクラスタに属する頂点に関する頂点データ23から成る。頂点データ23は各頂点毎に存在し、頂点座標データ及び必要な頂点属性データ(頂点法線データなど)から成る。

【0041】図2の(B)に示されるヘッダデータ10は、予め決められた頂点集合(=クラスタ)毎に定められるデータ構造であり、そのクラスタを構成する頂点の個数を頂点個数フィールド11に持ち、後述するクラスタを構成する差分データのデータ変換で利用するスケール値をスケールフィールド12に持ち、そのクラスタの基準点を示す基準点インデックスフィールド13を有する。この他に、必要に応じてそのクラスタで表される物体の光の反射特性や質感を表す属性などを示すクラスタ属性フィールドを持つことも可能である。

【0042】前記頂点データ23に含まれる頂点座標データは、図2の(C)に示される浮動小数型頂点データ14又は図2の(D)に示される整数差分データ19とさ

れる。

【0043】前記浮動小数型頂点データ14は、頂点座標を浮動小数型データ(浮動小数点数データとも称する)で表すデータ構造である。頂点のx成分フィールド15、y成分フィールド16、z成分フィールド17は頂点のx、y、z座標成分を夫々4バイトの浮動小数点数表現で持つ。浮動小数型頂点データ14は、例えば、後述する差分データ表現の基準となる点(基準点)や、浮動小数型データに変換された差分データを表すことが可能である。

【0044】前記整数差分データ19は、x差分値フィールド20、y差分値フィールド21、z差分値フィールド22を持つ。整数差分データ19は、例えば、頂点と前記基準点との相対的な位置関係を、基準点と頂点との差分ベクトルで表したときのx成分(dx)、y成分(dy)、z成分(dz)を例えば1バイトの整数値で表したデータである。

【0045】図1に示したクラスタの例において、頂点V3をこのクラスタの基準点とし、それ以外の頂点V1、V2、V4、V5を近傍頂点と呼ぶことにする。図1のクラスタに対応するクラスタのデータ(クラスタデータ)の例を図3に示す。図3の(A)に示される整数差分表現では、基準点V3のx、y、z座標値をそれぞれの浮動小数点数データ型で表し、近傍点V1、V2、V4、V5については整数差分データ型で表す。図3の(B)にはそれらの浮動小数点差分表現が示されている。図3の(A)の近傍点V1、V2、V4、V5のデータは、(B)の夫々浮動小数点データで表した基準頂点と近傍頂点との差分ベクトルのx成分、y成分、z成分を、それぞれ1バイト整数に変換した値を示している。たとえば、基準点V3に対する近傍頂点V1の差分ベクトルは図3の(B)に例示されるように「(x1-x3、y1-y3、z1-z3)」と表される。整数差分データ19はこの浮動小数値を図3の(A)に例示されるように、整数化したものである。近傍頂点V1の整数データ型での差分ベクトルは「(Int(x1-x3)、Int(y1-y3)、Int(z1-z3))」と表記している。この整数差分値「Int(x1-x3)、Int(y1-y3)、Int(z1-z3)」と頂点データとの関係は図1に例示されている。ここで「Int」という記号は浮動小数データ型から整数データ型に変換した値を意味するものである。この変換方法については後ほど詳しく述べる。

【0046】前記記録媒体2に格納される画像データの座標データ表現に必要なデータサイズは、前記近傍頂点の位置情報を整数データ型に圧縮した整数差分データで表すことにより、全ての座標点の座標データを浮動小数データ型で表現する場合に比べて、約4分の1に低減できる。

【0047】《整数差分データへの圧縮方法》図5のフ

ローチャートには差分データ表現形式の整数差分データの生成若しくは変換方法、即ちポリゴンデータ圧縮方法が例示される。これは図1に示すような浮動小数型座標データV1(x1, y1, z1)~V5(x5, y5, z5)をもつ頂点集合から、クラスタデータ24の各頂点データ23を作成する方法を示すものである。

【0048】ステップS1ではクラスタ化対象頂点集合の選択を行う。前述したように、クラスタは複雑な3次元物体をより単純な部分に分割したときの各部分に対応して使われる。これらクラスタの例としては、剛体的な特性、すなわち空間内の移動でその形を変えないような特性をもつ物体を構成する頂点の集まりなどがある。これらクラスタは画像データのモデル化段階で定義されているものであり、ここではそれらクラスタに属する頂点の座標値を読み込む。各座標値は浮動小数点型データで表されているとする。

【0049】ステップS2では選択されたクラスタを構成する頂点の集まりから整数差分表現の基準となる頂点(基準点)を選択する。基準点以外の点を近傍点と呼ぶ。基準点と近傍点との差分ベクタの長さが各近傍点毎に大きくばらつかないようにするために、クラスタに属する頂点群の「重心点」を求め、例えばその点の座標値に近い一つの頂点を基準点とする。即ち、ここで重心点のx座標値は、各頂点のx座標値の平均値、すなわち、x座標値の総和を頂点個数で割った値とする。同様に重心点のy座標値、z座標値は、それぞれ、各頂点のy座標値の平均、z座標値の平均とする。この重心点と各頂点の距離を計算し、重心点から最小の距離にある頂点をこのクラスタの基準点とする。基準点以外の点は近傍点と呼ぶことにする。

【0050】ステップS3では、基準点と近傍点との差分ベクタを整数化するためのスケール値を計算する。基準点と各近傍点の距離のうち、最大の値をLとおく。このLを例えば値127で除算した商をこのクラスタのスケール値(浮動小数点数)とし、Sで表すこととする。ここでは、1バイトの符号付き整数で差分値を0~127の値で表すためである。もし、整数値を表すビットサイズを変えたいならば、そのビットサイズで表現できる正の整数値をSとすればよい。このとき、整数差分データ19の各成分は当該ビットサイズに変更すればよい。

【0051】ステップS4では、スケール値から近傍点の整数差分を計算する。まず、各近傍点と基準点との差分ベクタを浮動小数型で表す。例えば図3の(B)の浮動小数差分表現では近傍点V1に対する差分ベクタは「(x1-x3, y1-x3, z1-z3)」と表されている。次にこの浮動小数型の差分ベクタの各x, y, z成分をステップS3で計算されたスケール値Sでそれぞれ除算し、四捨五入した商を整数値に変換する。この操作を「Int」という記号で表すとき、上記例の浮動小数型差分ベクタ「(x1-x3, y1-x3, z1-

z3)」は整数型差分ベクタ「(Int(x1-x3), Int(y1-x3), Int(z1-z3))」で近似表現できる。この操作を全ての近傍点に対して行う。図3の(A)の整数差分表現では近傍点V1, V2, V4, V5に対する整数差分データ19のx, y, z成分が示されている。

【0052】浮動小数点数から整数への変換方法については、公知でありここではその詳細な説明は省略するが、例えば、以下のような処理で実現できる。浮動小数点数の仮数Mの有効桁数Lを整数データの桁数に合わせて、仮数をM=1.b1b2b3...bLと変形する。指数EのゲタE0と前記浮動小数点数の仮数の有効桁数Lとの和の値E0+Lを取得し、E<E0+Lならば、E=E0+Lになるまで、仮数Mを1桁右シフトし、Eを+1し、E=E0+Lのときの値1b1b2b3...bLを整数の絶対値とする。浮動小数点数の符号が負の場合にはその絶対値を2の補数に変換する。これによって、浮動小数点数から整数への変換が完了される。

【0053】ステップS5では、ステップS4で得られた近傍点の整数差分データの各成分から整数差分データ19を構成し、このクラスタに対応するヘッダデータ10、基準点に対応する浮動小数点型頂点データ14、及び整数差分データ19を合わせてクラスタデータ24を構成する。

【0054】ステップS6では、未処理のクラスタがあるか判定し、未処理のものがあれば、そのクラスタを上記同様に処理する。

【0055】《浮動小数点差分データへの伸張》前記差分データ変換器4は整数差分データ19を浮動小数点型データへ変換する機能を有する。前記差分データ変換器4はまず、入力された整数差分データ19のx差分値フィールド20の値dX、y差分値フィールド21の値dY、z差分値フィールド22の値dZをそれぞれ4バイトサイズの単精度浮動小数点形式に変換する。

【0056】整数から浮動小数点数への変換方法については、公知でありここではその詳細な説明は省略するが、例えば、以下の処理で実現することが可能である。整数データの符号を浮動小数点数データの符号(S)とし、整数データの絶対値の2進表現(b0b1b2...bL)を求め、2進表現の桁数(L+1)が仮数の桁数になるようにする。前記絶対値の2進表現を正規化前の仮数(M=b0.b1b2...bL)とし、その2進表現の桁数(L)及び指数のゲタ(E0)から正規化前の指数(E=E0+L)を決定し、それらによって正規化されていない浮動小数点数が求められる。これを正規化することによって整数データの浮動小数点数への変換が完了される。

【0057】次に、浮動小数で表現された各差分値にスケールフィールド12のスケール値を乗算することにより浮動小数点型差分データに変換される。例えば、図3

の(A)における整数差分表現に示される各整数値は、図3の(B)における浮動小数差分表現のように4バイトの浮動小数値に変換される。すなわち、クラスタデータ10を構成する各頂点データ23のうち、近傍点に対応するものは、データ変換により整数差分データ19のデータ構造から、浮動小数点型差分データに変換される。ただし、ここでの x 、 y 、 z 成分の意味は、基準点と近傍点との差分を表すベクトルの各 x 、 y 、 z 成分である。以降のアフィン変換や透視変換などの処理では、ここで変換された浮動小数点差分データに対して演算が行なわれる。各頂点に関する浮動小数点差分型差分データのデータサイズは各成分毎に4バイトに伸張されている。

【0058】《アフィン変換》図6には伸張された差分データを用いるアフィン変換の一例が示される。図6の(B)に示される式1において、 W 、 V 、 P はそれぞれ3次元空間内のベクトルを表し、 M は3行3列の正方行列を表すとする。 $*$ は積を意味する。このとき、ベクトル V からベクトル W への変換をアフィン変換といい、アフィン変換は行列 M とベクトル P で決定される。ベクトル P は3次元空間内の平行移動と解釈される。図6の(B)における式2は式1のアフィン変換を成分で表現したものである。図6の(A)において点 $V1$ は前記アフィン変換によって点 $W1$ に変換され、点 $V2$ はアフィン変換によって点 $W2$ に変換されたものとする。また、ベクトル $dV2$ は点 $V1$ から点 $V2$ への相対ベクトルとし、その x 、 y 、 z 成分をそれぞれ、 $dx2$ 、 $dy2$ 、 $dz2$ で表す。またベクトル $dW2$ は点 $W1$ から点 $W2$ へ向かう相対ベクトルとする。

【0059】このとき、数学の線形性から、ベクトル $dW2$ はベクトル $dW1$ を行列 M で変換したものに過ぎない。すなわち、アフィン変換に含まれる平行移動の影響が相対ベクトルには表れない。この性質を利用して差分表現された頂点のクラスタデータを操作する。

【0060】今、図6の点 $V1$ はある頂点クラスタの基準点とし、点 $V2$ はそのクラスタの近傍点とする。このとき、差分データ変換器4で変換された近傍点 $V2$ の浮動小数点型差分データは、図7の(A)のアフィン変換前の座標差分表現のように示されている。

【0061】クラスタ内の基準点 $V1$ および近傍点 $V2$ に、それぞれ図6の(B)の式1、式2で表されるアフィン変換を適用してできる点をそれぞれ $W1$ 、 $W2$ とみると、 $W1$ は変換後クラスタ内の基準点、 $W2$ は変換後クラスタ内の近傍点となる。そのとき、変換後のクラスタデータは図7の(B)のアフィン変換後座標差分表現に示されるようになる。すなわち、アフィン変換に含まれる平行移動の影響が相対ベクトルには表れない性質から、近傍点に対しては、浮動小数点型差分データの x 、 y 、 z 成分から構成されるベクトルを単に図6の(B)の式1、式2の行列 M で変換すればよいことがわかる。

ただし、基準点 $V1$ に対しては、図6の(B)の式1、式2の行列 M での変換とベクトル P での平行移動が必要であることは言うまでもない。

【0062】クラスタに対して以上のようなアフィン変換を適用することにより、近傍点に対応する頂点のアフィン変換では平行移動の演算が不要となり演算量の軽減に寄与することができる。

【0063】《透視変換》図8には伸張された頂点差分データを利用した透視変換の手法が例示される。図8は3次元空間を z 軸に垂直な方向から見た図である。点 $V1$ と原点とを結ぶ直線と2次元透視平面30との交点を $W1$ で表す。このように点 $V1$ と原点を結ぶ直線と透視平面30との交点 $W1$ を求めることを透視変換といい、 $W1$ を透視点と呼ぶことにする。

【0064】透視点 $W1$ の x 、 y 、 z 座標値をそれぞれ、 $px1$ 、 $py1$ 、 $z0$ と表す。2次元透視平面上の点の座標値は方程式「 $z=z0$ 」を満たす平面なので、この平面上の点の z 座標値は常に $z0$ である。 $W1$ の x 、 y 座標値 $px1$ 、 $py1$ は幾何学の相似の関係より、図8(B)の式3および式4で計算できる。ここで式3および式4に含まれる値 A は式1により計算される。すなわち、透視点の座標値を計算するには、除算が1回と、乗算が2回あれば十分である。

【0065】点 $V2$ が点 $V1$ の十分近くにあれば、点 $V2$ の透視点 $W2$ の x 、 y 座標値を除算を使用しない近似式で計算できる。この近似式とは、数学の分野での微分可能関数「 $1/z$ 」を $z=z1$ の近くで「マクローリン級数展開」した級数(図8(B)の式10)を定数項と1次の項で打ち切ったものである。要するに、図8の式11のように $z0/(z1+dz2)$ に対して定数項と1次の項で打ち切ったマクローリン級数展開を施す。これにより、図8の点 $V2$ が十分点 $V1$ に近ければ、点 $V2$ の透視点 $W2$ の x 、 y 座標値はそれぞれ、図8の(B)の式5及び式6の $px2$ 、 $py2$ で近似できる。ここで、値 $dx2$ 、 $dy2$ 、 $dz2$ は $V1$ から $V2$ へ向かう相対ベクトルの x 、 y 、 z 成分であり、「 $x1+dx2$ 」、「 $y1+dy2$ 」、「 $z1+dz2$ 」はそれぞれ点 $V2$ の x 、 y 、 z 座標値を点 $V1$ の座標値と点 $V1$ から点 $V2$ へ向かう相対ベクトルの成分値で表したものである。

【0066】ここで、図8の(B)における式5及び式6に含まれる値 A 、 B は図8の(B)の式1及び式2により予め計算しておき、値「 $A-B*dz2$ 」も一度計算された値を式5、式6で再利用するものとすれば、透視点の座標値の近似計算には、実質的に乗算が3回と加算が3回あれば十分である。すなわち、毎回除算は不要となる(ここで記号「 $*$ 」は乗算を表すとする)。プロセッサによる除算に必要な演算サイクル数が乗算や加算に比べて大幅に大きいとき、上記のような近似計算を点 $V1$ の十分近くに存在する透視変換対象点に対して適用

することにより全体の演算処理時間が短縮される。例えば、浮動小数点データの除算に15サイクルかかり、浮動小数点データの乗算と加算にそれぞれ2サイクル掛かるとすれば、近似計算しない場合、除算が1回、乗算が2回、加算が2回で合計、23サイクルかかる。一方、近似計算すれば、乗算が3回と加算が3回で合計12サイクルで済む。

【0067】図9には浮動小数点差分データを用いた透視変換処理のフローチャートが示される。いま、図8の点V1はあるクラスタの基準点であり、点V2はそのクラスタの近傍点であるとする。

【0068】ステップS11で、除算を含む正規の透視変換により、先ず基準点に対応する透視点を計算する。すなわち、図8の(B)における式1、式3、式4で透視点の座標値を計算する。加えて以降の近似計算で繰返し再利用される値を式2で計算しておく。

【0069】ステップS12はこのクラスタに属する近傍点に対して除算を含まない近似計算を適用するかどうかを判定する。基準点V1のz座標値が座標原点に近すぎると近似による誤差が拡大するので、画像処理プログラムで予め決定されているある値cより小さいときは誤差が顕著に表れるのを防ぐためにステップS13に進む。そうでないときはステップS14に進む。

【0070】ステップS13では図8の(B)における式7、式8、式9により近傍点に対応する透視点のx、y座標値を計算する。dx、dy、dzはそれぞれ近傍点に対応する浮動小数型頂点データのx、y、z成分値である。

【0071】S14では近傍点に対応する透視点のx、y座標値の近似で共有される値「 $A-B \cdot dz^2$ 」を計算する。ここで、A、BはステップS11において図8の(B)における式1、式2を用いて計算された値であり、dz²は近傍点に対応する浮動小数型頂点データのz成分値である。

【0072】ステップS15では近傍点に対応する透視点のx座標値の近似を図8の(B)における式5で計算する。dx²は近傍点に対応する浮動小数型頂点データのx成分値である。

【0073】ステップS16では近傍点に対応する透視点のy座標値の近似を図8の(B)における式6で計算する。dy²は近傍点に対応する浮動小数型頂点データのy成分値である。

【0074】ステップS17で与えられたクラスタ内で未処理の近傍点があれば、ステップS15に進み、なければ当該クラスタに属する点の透視変換を終了する。

【0075】前記プロセッサ5が以上説明した「アフィン変換」及び「透視変換」を行う為の処理プログラムは、特に制限されないが、前記記録媒体2から読み込まれた前記画像処理プログラムに含まれている。

【0076】ここで、以上説明した整数差分データ構造

を採用することによる画像情報の圧縮、圧縮された画像情報の伸張、伸張して得られる浮動小数点差分データを利用する「アフィン変換」及び「透視変換」の夫々、により得られる作用効果を説明する。一般に、3次元画像のポリゴン表現においてポリゴン数を増やせば増やすほど、よりなめらかでより緻密な3次元画像を構成できるとされている。一方で、ポリゴン数を増やすということは頂点個数および必要なデータ量が増大することを意味し、また画像処理に必要な計算量が増大することを意味する。

【0077】ポリゴンを構成する頂点の位置情報が座標成分あたり4バイトの浮動小数点データで表されるモデルにおいて、上述のように、この位置情報を整数データ型に圧縮した整数差分データ19に変換することにより、画像データ格納時にはポリゴンの位置情報に必要なデータサイズを約4分の1に圧縮することができる。

【0078】また、伸長処理は各頂点データに対して、1バイト整数データを4バイト浮動小数点データにデータ型変換してスケール値(フィールド12のデータ)を乗算するので、複雑な伸長処理が不要である。

【0079】また、整数差分データ19を浮動小数点型頂点データに伸長したとき近傍点の各x、y、z成分は基準点に対する相対値であるという特徴を利用して、アフィン変換では平行移動の演算が不要となる場合があり、演算量の軽減に寄与することができる。

【0080】さらに、頂点数が増えるということは頂点間の距離が近づくことを意味しており、透視変換処理において、除算を乗算と加算の組み合わせで近似することが可能で、基準点と近傍点間の距離が近づくことにより誤差も小さく済むため、除算に必要なサイクル数が大きいプロセッサなどでは、画像の品質を落とさずに透視変換処理時間を全体的に短縮することができる。

【0081】《整数差分データの別の例》上記の説明では、図5のフローチャートに基づいて説明したように、クラスタの基準点としてポリゴンを構成する頂点の中から一つを選んだ。すなわち、ステップS2において、クラスタに属する頂点群の「重心点」を求め、その点の座標値に近い頂点を基準点とした。このとき、前記重心点そのものを基準点とすることも可能である。すなわち、図10の(A)に示すように点V1、V2、V3、V4、V5から成るクラスタの重心が点Pであるとき、クラスタを構成する各頂点V1、V2、V3、V4、V5を全て近傍点として、それらの整数差分データ19の内容を図10の(B)の整数差分表現で表すことができる。ここで、x0、y0、z0はそれぞれクラスタの重心点のx、y、z座標値とし、「Int」は前述と同様の意味を持つ。

【0082】こうしたときのクラスタデータ24における前述の例との相違点は、ヘッダデータ10において基準点となる頂点を指定するための基準点インデックスフ

フィールド13が不要になることである。基準点Pの位置情報を浮動小数型頂点データ14の形式で表し、データ構造としては、ヘッダデータの直後に配置すればよい。ここで基準点Pはポリゴンを構成する頂点ではないので、基準点Pに関しては頂点法線データ等の付加属性データは不要である。そして各頂点に対応する頂点データ23をデータ構造としては基準点Pに対応するデータの直後に順番に配置する。この順番は例えば、ストリップ表現で規定される頂点順序とすればよい。

【0083】アフィン変換においても、基準点Pと近傍点との関係は前述の例とほぼ同様である。相違点は、クラスタを構成する頂点が全て近傍点の役割を担い、基準点Pはポリゴンの頂点とはならないことである。透視変換においても、基準点Pと近傍点との関係は前述の例とほぼ同様である。違いは、クラスタを構成する頂点が全て近傍点の役割を担い、基準点Pは実際の透視平面上での表示対象とはならないことである。

【0084】基準点をこのようにする場合にも、前述の例と同様に、クラスタの各頂点の位置情報を整数データ型に圧縮した整数差分データに変換することにより、画像データ格納時にはポリゴンの位置情報に必要なデータサイズを約4分の1に圧縮することができる。

【0085】また、画像データを整数差分データに圧縮する処理において、基準点をクラスタを構成する頂点から選ぶ必要がないため、圧縮時間が前述の例に比較してさらに短縮できる。さらに基準点がクラスタデータ24上でいつも同じ位置（例えばヘッダデータ10の直後）に配置され、且つ頂点データが全て同じ差分データ構造をもつので、画像処理プログラム実行時にプロセッサなどによるデータアクセスが規則的に行え、この点においても処理時間の短縮を促進できる。

【0086】また、前述の例と同様にアフィン変換では平行移動の演算が不要となる場合があり演算量の軽減に寄与することができる。さらに、前述の例と同様に、透視変換処理において、除算を乗算と加算の組み合わせで近似することが可能で、基準点と近傍点間の距離が近づくことにより誤差も小さく済むため、除算に必要なサイクル数が大きいプロセッサなどでは、画像の品質を落とさずに全体の処理時間を短縮できる。

【0087】《浮動小数点差分データへの別の変換例》
前述の例では、プロセッサ5のアクセラレータなどとして位置付けられるハードウェアとしての差分データ変換器4により、整数差分データ19を浮動小数型差分データに変換しているが、別の実現方式として、差分データ変換器4を用いずに、プロセッサ5で実行される所定のデータロード命令に、上記変換機能をもたせることも可能である。

【0088】例えば、図4において差分データ変換器4を削除する。各クラスタデータ24は画像データの記録媒体2から一旦メモリ7にストアされているとする。こ

のストア処理はプロセッサ5が行ってもよいが、図示を省略するDMAC（ダイレクト・メモリ・アクセス・コントローラ）で行ってもよい。プロセッサ5は、画像処理プログラムに従って、頂点の位置データをプロセッサ5内のレジスタにロードする機能と整数データを浮動小数にフォーマット変換する機能を持つ所定のデータロード命令（FLOAD命令）により、メモリ上の整数差分データ19のx差分値フィールド20、y差分値フィールド21、z差分値フィールド22をそれぞれ浮動小数点数形式に変換して浮動小数点レジスタにロードできる。その後のアフィン変換や透視変換は前述の例と同様に、画像処理プログラムに従って行うことができ、前述の例と同じ効果を得ることができる。また、差分データ変換器4を用いずに、プロセッサ5で実行されるFLOAD命令に、上記変換機能をもたせることにより、画像データのデータ圧縮形式が多少変更されても、プログラムの変更でデータ伸長機能を柔軟に対応させることが可能となる。

【0089】図11には前記FLOAD命令をサポートするプロセッサの一例が示される。同図に示されるプロセッサ5は、特に制限されないが、32ビットRISC (Reduced Instruction Set Computer)アーキテクチャを有し、その命令セットには16ビット固定長の浮動小数点命令を含んでいる。ここで説明する例は、ゲーム機などのように3次元グラフィックスをサポートすることが必要な機器組み込み制御用途に有効である。

【0090】プロセッサ5は、命令制御回路42、整数ユニット43、浮動小数点ユニット44、命令キャッシュユニット45、及びデータキャッシュユニット46を有する。

【0091】命令制御回路42は命令アドレスバス48及び命令バス49を介して命令キャッシュユニット45に接続されている。また、整数ユニット43及び浮動小数点ユニット44はデータバス50を介してデータキャッシュユニット46に接続されている。データバス50を介するデータアクセスのためのアドレッシングは専ら整数ユニット43が行うようになっており、アドレスバス51は整数ユニット43からデータキャッシュユニット46に接続されている。

【0092】命令制御回路42は、プログラムの実行順序に従って命令キャッシュユニット45から命令をフェッチし、これを解釈して制御信号を生成し、前記整数ユニット43及び浮動小数点ユニット44等の動作を制御する。プログラムの実行順序は、図示を省略するプログラムカウンタの値や割り込み要求に基づいて決定される。命令アドレスは命令アドレスバス47から命令キャッシュユニット45に与えられ、その命令アドレスの命令が命令キャッシュユニット45から命令バス48を介して命令制御回路42に与えられる。

【0093】前記整数ユニット43は整数演算回路6

0、汎用レジスタファイル61、及び整列・拡張回路62を有する。整数演算回路60は算術論理演算器、算術演算器、及びシフト等を有し、整数データの算術演算、論理演算、そしてアドレス演算を可能にする。汎用レジスタファイル62は、特に制限されないが、32ビット長の汎用レジスタを複数個有し、バイト(8ビット)、ワード(16ビット)、ロングワード(32ビット)のデータレジスタ並びにアドレスレジスタとして利用される。前記データバス50は、特に制限されないが、64ビットである。前記整列・拡張回路62は、64ビット幅のデータバスに伝達されるデータと汎用レジスタとのビット位置をデータサイズなどに応じて整合させるアラライナ機能と、データを符号拡張又は論理値“0”拡張する拡張機能を有する。

【0094】尚、前記整数ユニット43と命令制御回路42の全部又は一部の機能を合せた回路構成を所謂CPUとして位置付けてよい。

【0095】前記浮動小数点ユニット44は、浮動小数点演算回路70、浮動小数点レジスタファイル71、及び変換回路72を有する。前記浮動小数点演算回路70は浮動小数点数データの積和演算を可能にする乗算器、加算器、正規化器等を有する。浮動小数点レジスタファイル71は複数個の32ビット浮動小数点レジスタを有する。浮動小数点レジスタは浮動小数点演算においてソースデータレジスタ及びディスティネーションデータレジスタ等に利用される。浮動小数点レジスタは前記変換回路72を介してデータバス50に接続されている。各浮動小数点レジスタは32ビットであり、単精度浮動小数点数データ毎に1個の浮動小数点レジスタが用いられる。倍精度浮動小数点数データには2個の浮動小数点レジスタがペアで割り当てられる。

【0096】前記変換回路72は、64ビット幅のデータバス50に伝達されるデータと浮動小数点レジスタとのビット位置をデータサイズなどに応じて整合させるアラライナ機能と、整数データを符号拡張又は論理値“0”拡張する拡張機能と、整数データを浮動小数点数データに変換する型変換機能、そしてその逆の変換を行う逆型変換機能を有する。それら機能の詳細については後述する。

【0097】前記データキャッシュユニット46及び命令キャッシュユニット45は、夫々図示を省略するキャッシュコントローラ及びキャッシュメモリを備えている。前記命令キャッシュユニット45及びデータキャッシュユニット46はデータ信号やコントロール信号を含むキャッシュバス57を介してバスコントローラ52に接続される。命令キャッシュユニット45におけるキャッシュミスヒット等に起因する外部アクセスのための命令アドレスは前記バスコントローラ52に与えられる。また、データキャッシュユニット46におけるキャッシュミスヒット等に起因する外部アクセスのためのデータ

アドレスは前記バスコントローラ52に与えられる。バスコントローラ52はそれら命令アドレス又はデータアドレスに従って、バスインタフェースバッファ53に外部バス54を介して結合される外部メモリ(図示を省略)などをアクセスするために、外部バスサイクルを起動する制御を行う。また、バスコントローラ52にはタイマやシリアルコミュニケーションインタフェースコントローラ等の周辺回路55が周辺バス56を介して接続されている。第1図に示されるプロセッサ5は、単結晶シリコンのような1個の半導体基板(半導体チップ)に形成されている。

【0098】前記プロセッサ5の命令セットは固定小数点転送命令、算術演算命令、論理演算命令、分岐命令、システム制御命令、浮動小数点命令などに大別される。前述のように、データアクセス及び命令フェッチのためのアドレッシング機能は全て整数ユニット43が負担する。従って、命令制御回路42は、デコードした命令が浮動小数点命令(浮動小数点ユニットを動作させる必要のある命令)である場合、整数ユニット43にはソース又はディスティネーションデータをアクセスするためのアドレッシング動作を制御し、浮動小数点ユニット44には演算制御などを指示する。

【0099】図12には前記変換回路72及び整列・拡張回路62の一例が示される。前記データキャッシュユニット46のデータ入出力ポートは64ビットであり、データアクセスにおいてバス50上には64ビットのデータが現れる。前記整列・拡張回路62はアラライナ62Aと拡張回路62Bを含む。前記変換回路72はアラライナ72A、型変換・逆変換回路72C及び拡張回路72Bを有する。図12の回路接続状態はデータキャッシュユニット46から汎用レジスタファイル61、浮動小数点レジスタファイル71へのデータロードを想定して図示してある。浮動小数点レジスタ71の値をデータキャッシュユニット46にストアするデータ転送方向の接続状態は図示を省略してある。

【0100】汎用レジスタファイル61にデータをロードするとき、前記アラライナ62Aは、データバス50から供給される64ビットデータ内の1バイトデータ、2バイトデータ、又は4バイトデータを32ビット出力中の対応する最下位ビットフィールドへシフトさせる。この32ビットを受ける拡張回路62Bは、入力32ビットの最下位の1バイト又は最下位から2バイトを32ビットへゼロ拡張(論理値“0”拡張)し、或いは符号拡張する。

【0101】汎用レジスタファイル61からデータキャッシュメモリ46にデータをストアするときは、特に図12には図示を省略しているが、アラライナ62Aが汎用レジスタの出力を受けて、汎用レジスタファイル61から供給される32ビットデータ内の1バイト又は2バイト、又は4バイトを64ビット出力中の対応する最下位

ビットフィールドヘシフトさせる。このとき拡張回路62Bはその64ビットを受けて、入力64ビット中の1バイト、2バイト又は4バイトをゼロ拡張し、或いは符号拡張してデータバス50に出力する。

【0102】データキャッシュメモリ46から浮動小数点レジスタファイル71にデータをロードするとき、アライナ72Aは、データバス50から入力する64ビットの内の1バイト、2バイト、又は4バイトを、出力64ビットの上位32ビットまたは下位32ビット中の対応する最下位ビットフィールドヘシフトさせる。この64ビットを受ける型変換・逆変換回路72Cは、入力64ビットの上位32ビット及び下位32ビット中の最下位1バイト又は最下位から2バイトの前記上位及び下位2つの整数データを各々単精度の浮動小数点数データへ並列的に変換可能にされている。拡張回路72Bは型変換・逆変換回路72Cの動作に代えて動作され、入力64ビットの上位32ビット整数データ及び下位32ビット整数データ中の最下位1バイト又は最下位から2バイトの上位側をゼロ拡張（論理値“0”拡張）或いは符号拡張し、整数データのまゝ、浮動小数点レジスタファイル71に転送する。

【0103】浮動小数点レジスタファイル71からデータキャッシュメモリ46にデータをストアするときは、特に図12には図示を省略しているが、型変換・逆変換回路72Cは、入力64ビットの上位32ビット浮動小数点数データ及び下位32ビット浮動小数点数データを、夫々1バイト又は2バイトの整数データに変換する。或いは、入力64ビットの倍精度浮動小数点数データを32ビットの整数データに変換する。変換された整数データは、アライナに64ビットのデータとして入力され、上位32ビット及び下位32ビットの対応するビットフィールドの下位側に夫々配置されてデータキャッシュユニット46に向けて出力される。

【0104】前記変換回路72及び整列・拡張回路62の動作は命令制御回路42による命令デコード結果に従って制御される。

【0105】図13にはプロセッサ1の浮動小数点ロード命令として“FLOAD, SW2S @Rm, FRn”が例示される。同図に示される浮動小数点ロード命令は、データキャッシュユニット46のキャッシュメモリから2バイトの整数データを変換回路72に与え、この変換回路72で単精度の浮動小数点数データに変換し、単精度の浮動小数点レジスタへロードする処理を指示する命令である。

【0106】図13に例示される浮動小数点ロード命令“FLOAD, SW2S @Rm, FRn”において、80は主オペコード（主オペレーションコード）フィールド、81はアドレッシングモードフィールド、82サブオペコード（サブオペレーションコード）フィールド、83はリザーブフィールド、84はソースオペラ

ンドサイズフィールド、85はベースレジスタ指定フィールド、86はデスティネーションレジスタサイズ指定フィールド、87はデスティネーションレジスタ指定フィールドである。

【0107】図13の例では、主オペコードフィールド80には浮動小数点ロードを示す主オペコードfieldが割り付けられ、アドレッシングモードフィールド81にはレジスタ間接を示すアドレッシングモードrriが割り付けられ、サブオペコードフィールド82には符号付き整数から単精度浮動小数点数への型変換を示すサブオペコードidfが割り付けられ、リザーブフィールド83はゼロフィールドとされ論理値“0”が割り付けられ、ソースオペランドサイズ指定フィールド84には2バイト（word）を示すサイズwが割り付けられ、ベースレジスタ指定フィールドにはベースレジスタとなる汎用レジスタの番号を示すベースレジスタ番号mが割り付けられ、デスティネーションレジスタサイズ指定フィールドにはデスティネーションレジスタとなる浮動小数点レジスタのサイズsが割り付けられ、デスティネーションレジスタフィールドにはデスティネーションレジスタ番号nが割り付けられている。

【0108】なお、ここではソースオペランドは2バイトとされるが、これは、各頂点成分毎に1バイトに圧縮された整数差分データは、説明の便宜上、上位側が“0”拡張されて2バイトにアライメントされているものと理解されたい。

【0109】図14には前記浮動小数点ロード命令“FLOAD, SW2S @R1, FR4”を実行したときのデータフローが例示される。図14の例では、アドレス0x0104（0xは16進数を意味する）番地に格納されている2バイト整数データ0x0003を浮動小数点レジスタ4番（FR4）へロードする命令“FLOAD, SW2S @R1, FR4”を一例とする。

【0110】汎用レジスタ1番（R1）の内容はソースオペランドのアドレス0x0104とする。0x0100番地には2バイトの整数0x0001、0x0102番地には2バイトの整数0x0002、0x0106番地には2バイトの整数0x0004がそれぞれ格納されている。

【0111】第1ステップとして、先ず、浮動小数点ロード命令“FLOAD, SW2S @R1, FR4”が発行されると、汎用レジスタR1の内容0x0104をアドレスとしてデータキャッシュユニット46がアクセスされ、0x0104の最下位4ビットをゼロとした0x0100番地からの8バイト（64ビット）の整数データ0x0001 0002 0003 0004が読み出される。

【0112】第2ステップとして、アライナ72Aはデータキャッシュユニット46から読み出された8バイトのデータに対する整列制御の為に、前記アドレス0x0

104の最下位ビット4の値(32ビット)を入力データ8バイトに関するオフセットとし(offset 4)、命令で指定されるアクセスサイズ2(16ビット)をサイズとし(size 2)、出力64ビットの上位又は下位32ビットのどちらへシフトするかを示す出力位置指ビットとしてデスティネーションレジスタ(FR4)のレジスタ番号4の最下位ビットの値である0をアップ/ローとし(up/low 0)とする制御信号を入力する。これにより、入力データ8バイト中のオフセット4バイト(32ビット)からの下位側2バイト0x0003を出力データ8バイト中のオフセット4から上位側へ2バイトシフトさせ、その他の出力値は対応するビット位置の入力値をそのままにして、8バイトデータ列0x0001 0003 0003 0004をアライナ72Aから出力する。

【0113】第3ステップとして、データ型変換・逆変換回路72Cはアライナ72Aの出力である8バイト、および変換の方法として符号付き2バイト整数を32ビット単精度浮動小数点数に変換することを意味する制御信号を前記命令制御回路42から入力し、入力8バイト中の上位4バイト及び下位4バイトの各最下位ビットを基準とした2つの2バイト0x0003、0x0004を2つの符号付き2バイト整数とし、各々を32ビット単精度浮動小数点数0x4040 0000、0x4080 0000に変換し、2つの32ビット単精度浮動小数点数である8バイトを出力する。

【0114】第4ステップとして、最後に型変換・逆変換回路72Cが出力する8バイトの上位4バイト0x4040 0000が浮動小数点レジスタファイル内の単精度浮動小数点レジスタ4番(FR4)へ書き込まれる。

【0115】以上の手順によって、0x0104番地の2バイト整数0x0003が32ビット単精度浮動小数点数データ0x40400000(3.0)に変換されて浮動小数点レジスタFR4へ書き込まれる。

【0116】浮動小数点レジスタFR4にロードされたデータには前記スケール値が乗算されて浮動小数点差分データとされ、これがアフィン変換などの画像処理に供される。

【0117】前記第3ステップにおいて整数データを浮動小数点数データに変換する手法には公知の方法を採用し、これを型変換・逆変換回路の変換アルゴリズム若しくは変換論理に反映しておけばよい。

【0118】前記FLOAD命令は、整数データのビット長を指示するビット長情報フィールド84を有している。整数から浮動小数点への型変換では整数データのビット長と所定の浮動小数点フォーマットの仮数のビット長との相違に応じたビット長拡張処理が行われるが、それに必要な整数データのビット長情報を、前記FLOAD命令のビット長情報フィールド84のデコード結果か

ら得られるから、ビット長の異なる整数データが混在する場合も、1命令で前記型変換を伴う処理を行うことができる。処理対象とする整数データのビット長情報を制御レジスタで指定する場合に比べ、処理対象とする整数データのビット長が変わっても、その都度、余計なレジスタアクセス動作を行わずに済み、データ処理能率も向上する。

【0119】《情報処理ネットワーク》図15には図11のプロセッサ(MPU)5を採用した情報処理装置(コンピュータ装置とも称する)を含む情報処理ネットワークの一例が示される。

【0120】図15に示される情報処理ネットワークはLAN(ローカル・エリア・ネットワーク)、インターネットなどのWAN(ワイド・エリア・ネットワーク)、無線通信ネットワークなどのシステムであり、94で示されるものがそのシステムにおける光ファイバ、ISDN回線、又は無線回線などの伝送媒体を意味している。伝送媒体94には、特に制限されないが、ホストコンピュータ装置93、ルータやターミナルアダプタ等の通信アダプタ95、96、97を介して代表的に示された端末コンピュータ装置90、91、92が接続されている。

【0121】端末コンピュータ装置90は、特に制限されないが、前記プロセッサ(MPU)5を有し、表示コントローラ(DISPC)103、ネットワークコントローラ(NETC)104、及びメインメモリ(DRAM)105が接続され、また、プロセッサ5の前記周辺回路55に接続されたフロッピーディスクコントローラ(FDC)100、キーボードコントローラ(KEYC)101、及びインテグレートッド・デバイス・エレクトロニクス・コントローラ(IDEC)102が設けられている。DISPC103はビデオRAM(VRAM)111に描画制御を行い、描画した表示データをディスプレイ(DISPLAY)110に表示制御する。NETC104は通信アダプタ95に接続され、送受信情報のバッファリング及び通信プロトコル制御等を行う。DRAM105はプロセッサ1のプログラム領域及びワーク領域などに利用される。前記FDC100にはフロッピーディスクドライブ装置106が接続され、前記記録媒体2の一例であるフロッピーディスク120から情報を読み取り、また、情報を書込む。KEYC101にはキーボード107が接続される。IDEC102にはハードディスクドライブ装置(HDD)108、CD-ROMドライブ装置(CDRD)109が接続される。HDD108は記録媒体2の別の例である磁気ディスクを有する。CDRD109は記録媒体2の更に別の例であるCD-ROM121を有する。尚、その他の端末コンピュータ装置91、92も上記同様に構成される。

【0122】例えば前記端末コンピュータ装置90を用い、前記クラスタデータ24によって特定される画像デ

ータを伸張し、これに対して前記アフィン変換や透視変換等の3次元グラフィックス処理を行う場合、そのためのプログラムは、例えばフロッピーディスク120やCD-ROM121からハードディスクドライブ装置108にインストールされる。このとき、フロッピーディスク120やCD-ROM121には前記伸張、アフィン変換、及び透視変換等の画像処理プログラムが予め記録されている。端末コンピュータ装置のセットメカがそのプログラムをハードディスクドライブ装置にプリインストールして提供する場合もある。

【0123】プロセッサ5はインストールされたプログラムを実行するとき、そのプログラムをDRAM105にロードし、DRAM105から順次命令をフェッチして実行する。尚、CD-ROM121に格納されているプログラムの一部を直接CD-ROMから取り出して実行することも可能である。

【0124】これにより、前記端末コンピュータ装置90は、フロッピーディスク120等を介して前記プログラムをインストールでき、或いはハードディスクドライブ装置108等から直接前記プログラムを実行できる。

【0125】また、端末コンピュータ装置90はホストコンピュータ装置93から前記プログラムをダウンロードすることができる。即ち、ホストコンピュータ装置93は、例えば圧縮された前記プログラムをハードディスク装置などに保有している。端末コンピュータ装置90がホストコンピュータ装置93と通信を確立した後、端末コンピュータ装置90がそのプログラムを指定してダウンロードを指示することにより、前記プログラムが伝送媒体94に伝送されて、端末コンピュータ装置90のハードディスクドライブ装置108にダウンロードされる。ダウンロードされたプログラムは、その後、伸張されて、所定のプログラム格納エリアにインストールされる。これによって端末コンピュータ装置90には、前記アフィン変換や透視変換などに3次元グラフィックス処理を行う機能が実現される。

【0126】前記クラスタデータに対してもCD-ROMディスク121などの記録媒体2又は伝送媒体94を介してコンピュータ装置90、91、92に取り込むことが可能である。

【0127】前記整数差分データによって圧縮されたクラスタデータを伸張し、伸張された浮動小数点差分データに対し前記アフィン変換及び透視変換を行うための、データ処理プログラムを記録媒体2に記録して提供することにより、そのようなクラスタデータを用いた前記アフィン変換及び透視変換の処理を容易に実現できるようになる。

【0128】また、図1で説明した整数差分データを含むクラスタデータを記録媒体2(120、121)に記録して提供すれば、そのデータを用いて画像処理が行なわれることにより、圧縮データの伸長がプロセッサ処理

性能に対して大きな負担をかけず、また、記録媒体及びデータ転送のコストを減少させる。

【0129】以上本発明者によってなされた発明を実施形態に基づいて具体的に説明したが、本発明はそれに限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは言うまでもない。

【0130】例えば、前記整数差分データの伸張、伸張された浮動小数点差分データを用いる前記アフィン変換及び透視変換の為のデータ処理は、汎用的な制御にも用いられるプロセッサが行う場合に限定されず、グラフィックプロセッサなどの専用プロセッサ若しくはアクセラレータで行うことも可能である。また、プロセッサの具体的な構成も図11に限定されず、適宜変更可能である。また、圧縮されたクラスタデータを伸張し、伸張された浮動小数点差分データに対し前記アフィン変換及び透視変換を行うためのプログラムは、圧縮されたクラスタデータを記録した記録媒体とは別の単数又は複数の記録媒体に記録して提供してよい。

【0131】

【発明の効果】本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば下記の通りである。

【0132】クラスタの頂点座標と基準点との差分データを用いてポリゴンを表現するデータ形式を採用するから、画像データのサイズを小さくでき、これを記録する記録媒体の容量を圧迫せず、また、その画像データの転送時間短縮にも寄与する。

【0133】圧縮方法は、ポリゴン頂点の浮動小数点で表現される位置情報を整数値に変換する形式であるから、複雑な圧縮装置を必要とせず、ソフトウェアでも容易に対処することができる。

【0134】ポリゴン頂点の位置情報を予め定められたサイズの整数値で表現できるため、圧縮後のデータサイズの圧縮度も容易に調整可能である。また、圧縮データが規則的な構造を持つことにより、簡易なデータ伸長手段で伸長できる。更に、圧縮データが規則的な構造を持つことにより、メモリアクセス機能と整数型から浮動小数型へのデータ変換機能とを持つ命令を実行できるプロセッサで容易に伸長することができる。

【0135】画像圧縮の基準点を、各頂点座標の重心に近い頂点、或いは重心そのものとすることができるため、基準点と頂点との差分データの値を小さくすることが容易であり、複雑な圧縮計算が不要になる。

【0136】伸長されたクラスタを構成する頂点のアフィン変換は、各頂点に対応する差分ベクトルに対して、アフィン変換における線形変換を適用するため、アフィン変換の演算処理を軽減することが可能になる。

【0137】クラスタに属する各頂点の透視変換において基準点に対応する値から乗算と加算の組み合わせで近似することにより、透視変換で必要となる除算処理を軽

減できる。これは除算完了に必要な時間が、乗算や加算より大きいようなプロセッサで有効である。

【0138】透視変換の近似計算の適用対象を選択可能にすることにより、精度が重要な頂点と、近似処理で十分な頂点を選択して、描画品質を劣化させずに高速処理が可能である。

【0139】整数差分データを伸張するプログラム、或いは、伸張された浮動小数点差分データを用いる前記アフィン変換及び透視変換の為のデータ処理プログラムを、記録媒体に記録して提供することにより、そのようなクラスタデータを用いた前記アフィン変換及び透視変換の処理を容易に実現できるようになる。

【0140】また、整数差分データを含むクラスタデータを記録媒体に記録して提供すれば、そのデータを用いて画像処理が行なわれることにより、圧縮データの伸長がプロセッサの処理におきな負担をかけず、また、記録媒体及びデータ転送のコストを減少されることができ

る。

【図面の簡単な説明】

【図1】3次元画像を構成する頂点集合としてのクラスタ及び整数座分データの意義を示す説明図である。

【図2】差分データによって圧縮された頂点データ等によって構成されるクラスタデータの一例を示す説明図である。

【図3】クラスタ内頂点座標に対するデータ御差分表現例示する説明図である。

【図4】図1のクラスタデータを利用する画像処理システムの一例を示すブロック図である。

【図5】整数差分データへの圧縮方法の一例を示すフローチャートである。

【図6】伸張された浮動小数点差分データを利用したアフィン変換方法を例示する説明図である。

【図7】図6のアフィン変換前の座標差分表現による頂

点データとアフィン変換後の座標差分表現による頂点データとを例示する説明図である。

【図8】伸張された浮動小数点差分データを利用した透視変換方法を例示する説明図である。

【図9】伸張された浮動小数点差分データを利用した透視変換方法を例示するフローチャートである。

【図10】クラスタに属する頂点群の重心点を基準点として整数差分データを生成する場合の一例を示す説明図である。

【図11】FLOAD命令をサポートするプロセッサの一例を示すブロック図である。

【図12】図11のプロセッサが備える変換回路及び整列・拡張回路の一例を示すブロック図である。

【図13】FLOAD命令の命令フォーマットを例示する説明図である。

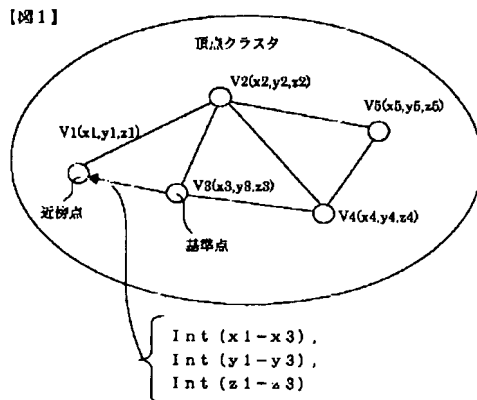
【図14】FLOAD命令を実行したときのデータフローを例示する説明図である。

【図15】情報処理ネットワークの一例を示すブロック図である。

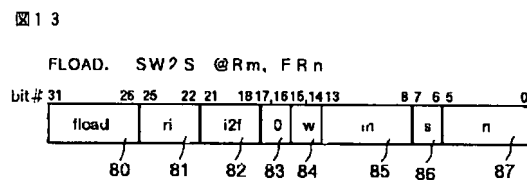
【符号の説明】

- 1 画像処理システム
- 2 記録媒体
- 4 差分データ変換器
- 5 プロセッサ
- 7 メモリ
- 10 ヘッダデータ
- 14 浮動小数点型頂点データ
- 15 整数差分データ
- 23 頂点データ
- 24 クラスタデータ
- 46 データキャッシュユニット
- 71 浮動小数点レジスタファイル
- 72 変換回路

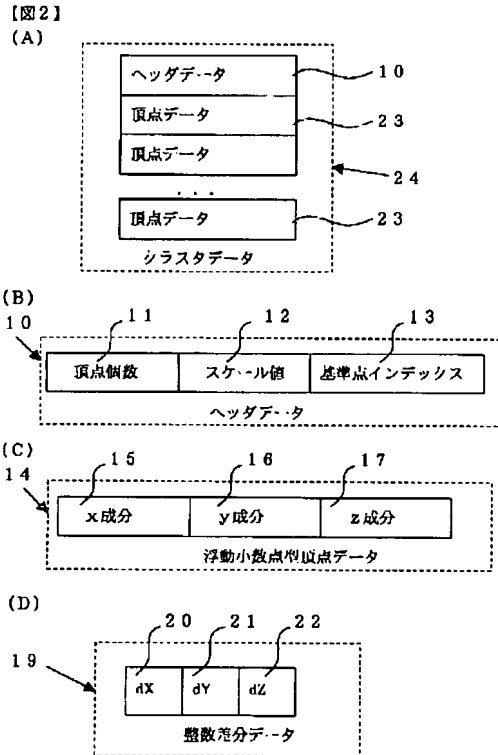
【図1】



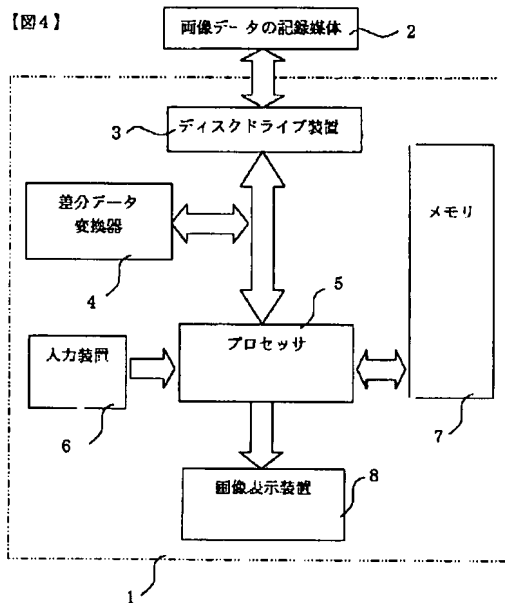
【図13】



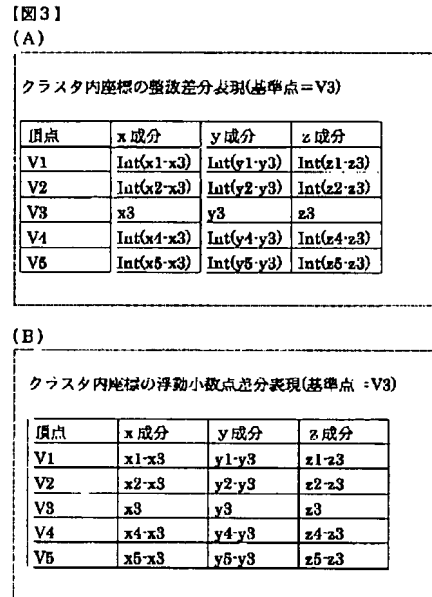
【図2】



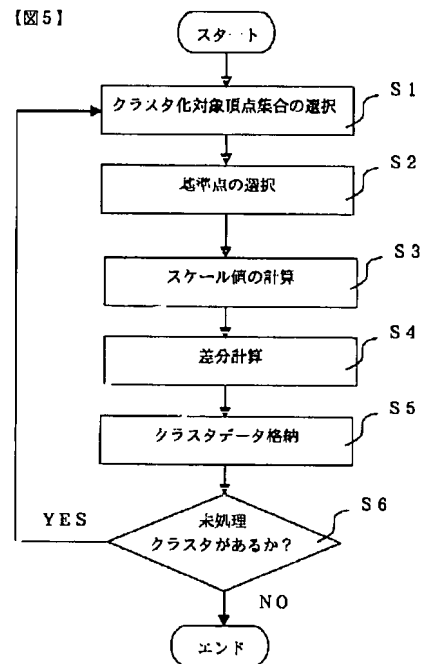
【図4】



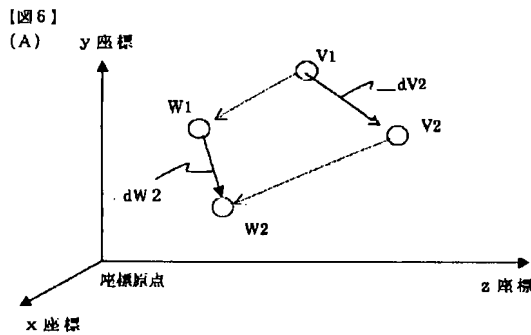
【図3】



【図5】



【図6】



(B)

アフィン変換: $W = M \cdot V + P$... (式1)

$$\begin{pmatrix} tx \\ ty \\ tz \end{pmatrix} = \begin{pmatrix} m11 & m12 & m13 \\ m21 & m22 & m23 \\ m31 & m32 & m33 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad \dots (式2)$$

【図7】

【図7】
(A)

アフィン変換前座標差分表現(基準点=V1)

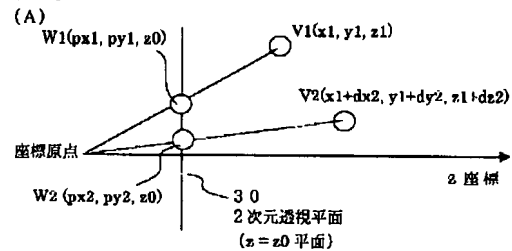
頂点	x 成分	y 成分	z 成分
V1	V1 の x 成分	V1 の y 成分	V1 の z 成分
V2	dV2 の x 成分	dV2 の y 成分	dV2 の z 成分

(B)

アフィン変換後座標差分表現(基準点=W1)

頂点	x 成分	y 成分	z 成分
W1	$M \cdot V1 + P$ の x 成分	$M \cdot V1 + P$ の y 成分	$M \cdot V1 + P$ の z 成分
W2	$M \cdot dV2$ の x 成分	$M \cdot dV2$ の y 成分	$M \cdot dV2$ の z 成分

【図8】

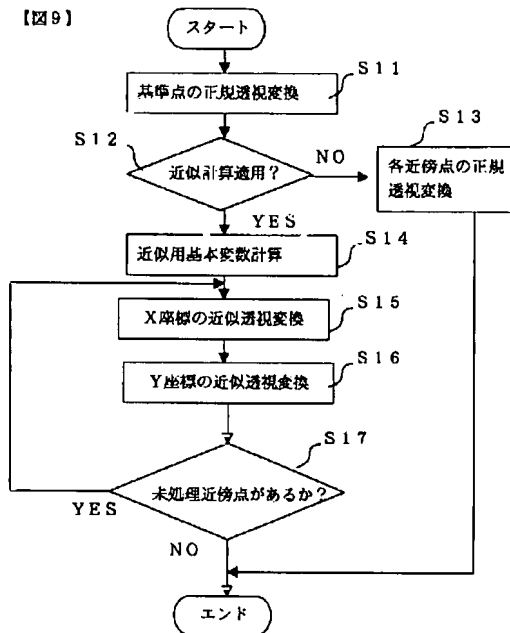
【図8】
(A)

(B)

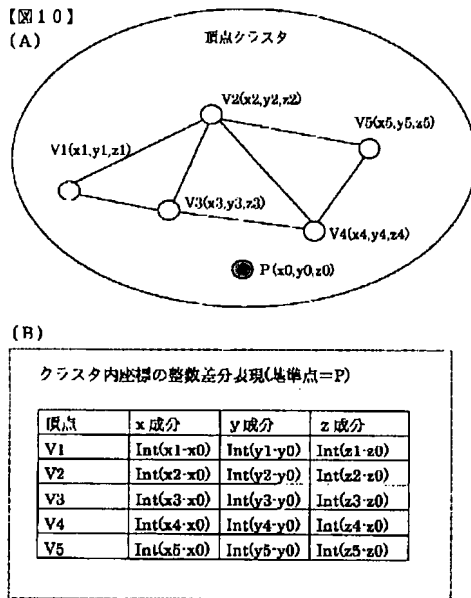
$$\begin{aligned} A &= z_0 / z_1 && \text{式1} \\ B &= z_0 / (z_1 * z_1) && \text{式2} \\ px1 &= A * x_1 && \text{式3} \\ py1 &= A * y_1 && \text{式4} \\ px2 &= (A - B * dz_2) * (x_1 + dx_2) && \text{式5} \\ py2 &= (A - B * dz_2) * (y_1 + dy_2) && \text{式6} \\ D &= z_0 / (z_1 + dz_2) && \text{式7} \\ px &= D * (x_1 + dx) && \text{式8} \\ py &= D * (y_1 + dy) && \text{式9} \\ \frac{1}{z + dz} &= \frac{1}{z} - \frac{1}{z^2} dz + \frac{1}{z^3} dz^2 - \frac{1}{z^4} dz^3 + \dots && \text{式10} \\ \frac{z_0}{z_1 + dz_2} &= \frac{z_0}{z_1} - \frac{z_0}{z_1^2} dz_2 + \dots && \text{式11} \end{aligned}$$

【図9】

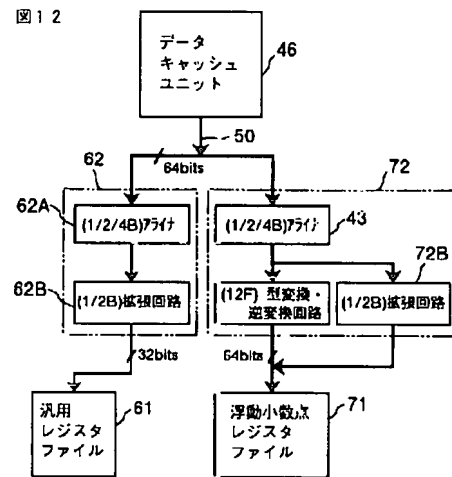
【図9】



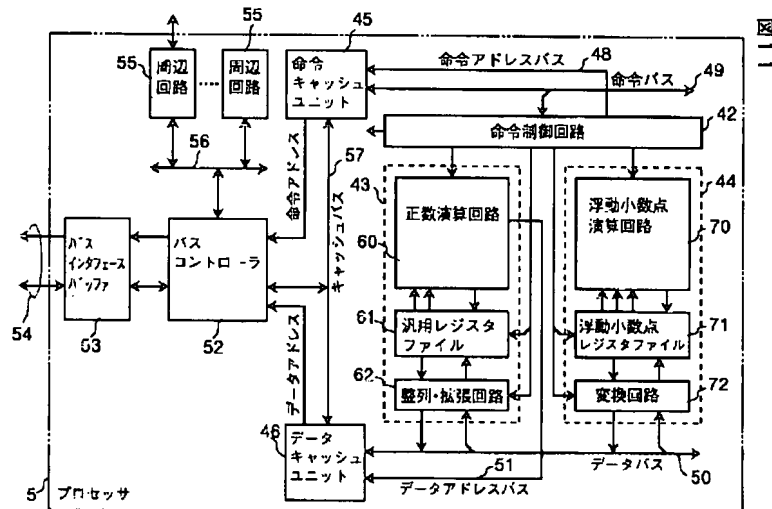
【図10】



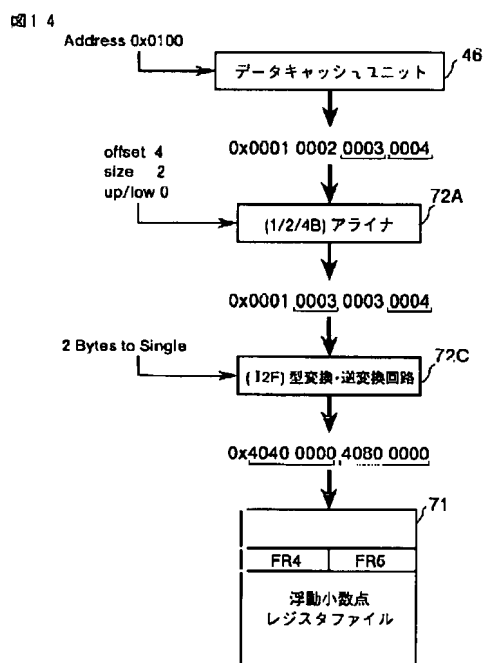
【図12】



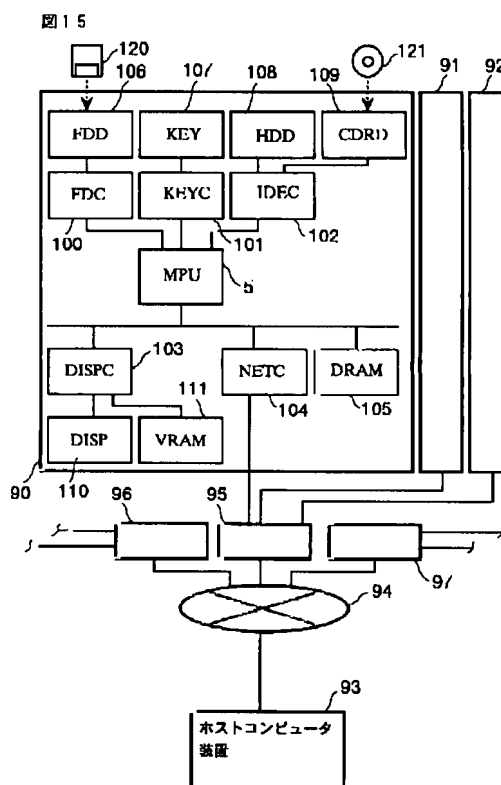
【図11】



【図14】



【図15】



フロントページの続き

(72)発明者 山田 孔司
東京都小平市上水本町五丁目20番1号 株
式会社日立製作所半導体グループ内

Fターム(参考) 5B050 AA08 BA10 EA10 EA27 FA02
FA05 FA06
5B057 BA28 CA13 CG10 CH08 CH11
DC06